Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 222 (2008) 128–143
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Methods for the rapid solution of the pricing PIDEs in exponential and Merton models

Anita Mayo

Baruch College, CUNY, New York, NY 10010, United States

Received 30 May 2006; received in revised form 12 November 2006

Abstract

The pricing equations for options on assets that follow jump-diffusion processes contain integrals in addition to the usual differential terms. These integrals usually make such equations expensive to solve numerically. Although Fast Fourier Transform methods can be used to to evaluate the integrals at all mesh points simultaneously, they are costly since the computational region must be extended in order to avoid problems with wrap around. Other numerical difficulties arise when the density function for the jump size is not smooth, as in the Kou double exponential model. We present new solution methods which are based on the fact that even when the problems contain time-dependent parameters the integrals often satisfy easily solved ordinary or parabolic partial differential equations. In particular, we show that by using the operator splitting method proposed by Andersen and Andreasen it is possible to reduce the solution of the pricing equation in the Kou and similar models to a sequence of ordinary differential equations at each time step. We discuss the methods and present results of numerical experiments.

MSC: 65M06; 91B70; 91B26

Keywords: Jump-diffusion process; Option pricing; Differential equations

1. Introduction

In response to the difficulties encountered when pure-diffusion models are used to price options on assets whose prices move rapidly, a variety of jump-diffusion models have been proposed. One of the most common models is due to Merton [13]. In his model the process the asset follows is the sum of a jump process with lognormally distributed jumps and the usual Weiner process. It is an attractive model since there is an analytic formula for pricing a European option in this model when its Black–Scholes price option is known. Furthermore, reasonable volatility smiles can be generated by carefully choosing the parameters of the jump process. Alternate models where the process is a pure jump process [11] have also been proposed. Other models which combine a local volatility function approach with jump processes have been used for pricing options [2]. More recently Kou [10] suggested a model with double exponential distribution for the log jump size. One reason for the model's popularity is its analytic tractability. Indeed, analytic formulas for many of the commonly used path-dependent options, such as lookbacks, barriers and perpetual

E-mail addresses: anita_mayo@baruch.cuny.edu, amayo@courant.nyu.edu.

0377-0427/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2007.10.017

Americans are known. Also, an extension of the Barone-Adesi and Whaley method can be used to approximate the values of regular American options in this model.

Although some analytic approximations are available for pricing options in these models it is still often necessary to use numerical methods, especially for most path-dependent options. These in turn require the solution of partial integro-differential equations (PIDEs). That is, in addition to (possibly degenerate) differential operators, the pricing equations contain a nonlocal integral term.

A broad class of methods have been used to numerically solve these equations. In particular, Amin [1] used a multinomial tree based approach. d'Halluin, Forsyth, Labahn and Vetzal [7–9] developed implicit methods for evaluating barrier options, American and Asian American options. They also showed that if Crank Nicolson discretizations of the differential and integral terms are used the scheme is unconditionally strictly stable if the coefficients are constant and a log spaced grid is used. In addition, they presented a rapidly convergent Picard iteration scheme for solving the discretized equations obtained when pricing options in the Merton model. Andersen and Andreasen derived a forward equation for European call options and applied it to calibration problems. They also presented a second-order accurate unconditionally stable operator splitting (ADI) method for pricing options which do not require iterative solution of an algebraic equation at each time step. These partial integro-differential equations have also been solved by many others. See for example, [18,12].

Because of the integral in the PIDE these numerical methods have, however, proven relatively expensive. Although the integral must be approximated at each point of the mesh used for discretizing the differential terms, it is possible to reduce the expense of computing the integral by making an exponential change of variables. It turns out this converts the integral into a correlation integral which can be evaluated at all the mesh points simultaneously using the Fast Fourier Transform [17,16,2].

However, since the integrals are not periodic it is usually necessary to extend the computational region significantly to avoid problems with "wrap around". Enlarging the grid increases the cost of the calculation, especially in higher dimensions. Fourier methods may also be slow to converge if the probability density of the jump size is not continuous (the Gibbs phenomenon), as it is in the Kou double exponential model. Moreover, for maximum efficiency they require uniform grids and the number of mesh points to be a power of two.

In this paper we present more efficient methods for solving the pricing PIDEs in exponential and Merton-type models. We first give a method for pricing options in the Kou double exponential model. The method uses the same operator splitting technique that Andersen and Andreasen used when pricing options in the Merton model. In their method they use a Crank Nicolson discretization of the PIDE. In most fully implicit methods for solving the pricing PIDE the equations that result at each time step are solved iteratively. By using the Andersen and Andreasen splitting in the Kou model, however, one obtains two coupled equations to solve at each time step which easily admit direct solution. Andersen and Andreasen discretized the differential parts of the equations using a standard method, and they evaluated the integral terms by Fast Fourier methods after carefully extending the solution to avoid wrap around effects.

We modify their scheme in several ways. First, we use a much faster way of evaluating the jump integral. We first note that after a linear change of variables the integrals in the Kou double exponential model separate. That is, at any point x the integral is the product of an exponential function of x and an integral where x only appears as the lower or upper limit of integration. This allows us to evaluate the convolution integral at all n points of the grid using only Cn operations where it is normally less than 3.

We also solve the equations in a different manner. It turns out that the fact that the kernel is separable allows us to convert the equations into linear boundary value problems (ordinary differential equations). If the parameters of the jump process are constant the ODE we obtain has constant coefficients. In any case, it is easily solved. In fact, this means that we can find option prices in the Kou model more rapidly than in Mertons model. We also note that we can extend our technique for evaluating the correlation integrals in the Kou model to evaluating the integrals in other similar models.

We also present a rapid method for evaluating the integrals in the Merton and related models. The method uses the fact that at any (physical) time the correlation integral is a translated solution of the heat equation. The heat equation is defined on the infinite line, and its initial values are equal to the solution of the PIDE at that time. The (nonphysical) evaluation time τ of the solution of the heat equation is equal to half the variance of the Poisson process, and the amount of the translation of the solution is equal to the expected value of the Poisson process. (When solving the pricing PIDE we either know the solution of the PIDE at the previous time step and want to evaluate the jump

integral using those values, or are using an iterative method to solve the PIDE and have an estimate for the solution of the PIDE at the current time.) In any case, we can evaluate the correlation integral by solving the heat equation numerically using a method that is fourth-order accurate in the log asset price and in τ. The method only requires 3 operations per point per time step [14], and since the variance of the Poisson process is usually small we found that by taking few steps in τ we could often evaluate the correlation integral to 5 digits of accuracy. Fourier methods require $Cn \log n$ operations where typically C is 10. This, and the fact that the region must be extended when using Fourier methods, mean that our methods can be much faster. In fact, numerical experiments indicate that they can be more than an order of magnitude faster. Our method also extends to more general related density functions. For example, we can similarly use differential methods to evaluate integrals when the density is the product of a polynomial with a Gaussian, or the product of a polynomial, a Gaussian and an exponential.

Another problem with using Fourier methods to evaluate the convolution integral in Merton-type models is that if the variance of the jump process is large the kernel of the integral is not rapidly decaying. This means that the integration region around any point must be large, which adds to the expense of the calculation. In contrast, boundary conditions for evaluating the jump integral by solving the heat equation are easily determined. Even ignoring the cost of the calculation, in our numerical experiments we found that for a given mesh width we could generally achieve greater accuracy by solving the heat equation with a small number of steps in τ than by computing the integral directly by quadrature. This was independent of the use of the FFT to perform the summation when evaluating the integral.

As for solving the entire pricing PIDE, we used the Picard iteration method studied in [7]. They gave necessary requirements for the iteration to converge, and our scheme satisfies these requirements if the method used to solve the heat equation is monotone. It turns out that our method is monotone if the step size in τ is sufficiently small compared to the square of the space step. However, numerical experiments indicate that our scheme may still converge even when this is not the case, although we have not proved this, nor have we proved that the our scheme converges to the viscosity solution. (See [4] for a a numerical scheme and a proof that it converges to a viscosity solution.)

Our evaluation methods can be used in higher dimensions. For example, the correlation integrals in Merton-type models and exponential models can also be evaluated as solutions of two and three-dimensional differential equations. Also, the density function in the Variance Gamma model is the fundamental solution of a Helmholtz equation in three dimensions. We note that our method cannot be used for pricing with general nonparametric jump density functions, although such densities can be approximated by piecewise exponential functions. (Cont and Tankov [5] have results on nonparametric calibration of jump-diffusion option pricing models). However, our method can be used when the asset follows a jump-diffusion process with (nonconstant) deterministic volatility function. Andersen and Andreasen claim that if such a model is used, then one obtains "reasonable and stable" deterministic volatility functions, without the "extreme short term variation typical of the pure-diffusion approach".

Finally, we note that when solving a forward problem Carr et al. [3] have used the fact that the double exponential density in the Kou model satisfies an ordinary differential equation. (They solve for the coefficient analytically, while we regard it as given.)

The rest of this paper is organized as follows. In Section 2 we present the pricing equations in jump-diffusion models and describe the standard numerical methods for solving them, in Section 3 we present our method for solving the PIDE in the Kou and related models, in Section 4 we give our method for evaluating the jump integrals in Merton-type models, and in the last section we provide results of numerical tests.

2. The basic equations and discretizations

In a jump-diffusion model the asset price motion is given by a process of the form

$$\frac{\mathrm{d}S}{S} = \nu \mathrm{d}t + \sigma \mathrm{d}z + (\eta - 1)\mathrm{d}q$$

where ν is the drift rate, σ is the volatility of the Brownian part of the process, and $\mathrm{d}q$ is a Poisson process. Here $\mathrm{d}q = 0$ with probability $1 - \lambda \mathrm{d}t$, $\mathrm{d}q = 1$ with probability $\lambda \mathrm{d}t$, where λ is the Poisson arrival intensity, and $\eta - 1$ is an impulse function giving a jump from S to $S\eta$. The average relative jump size, $E(\eta - 1)$ is denoted by κ. The Poisson process $\mathrm{d}q$ is assumed to be independent of the Wiener process $\mathrm{d}z$.

Merton [13] showed that with the above assumptions the value of a contingent claim $V(S,\tau)$ depending on the asset price S and time τ satisfies the following PIDE:

$$V_t = \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - (r+\lambda)\, V + \lambda \int_0^\infty V(S\eta) g(\eta) \mathrm{d}\eta \tag{2.1}$$

where $t = T - \tau$ is the time till expiration at T, r is the risk free interest rate, and $g(\eta)$ is the probability density function of the jump size η.

We first note that the integrals can be written as correlation integrals. That is, by making the exponential changes of variables $S = \mathrm{e}^x$, $\eta = \mathrm{e}^y$ the integral in (2.1) can be written as

$$IV(x,t) = \int_{-\infty}^{\infty} \tilde{V}(x+y,t) f(y) \mathrm{d}y \tag{2.2}$$

where $f(y) = g(\mathrm{e}^y)\mathrm{e}^y$ and $\tilde{V}(x+y,t) = V(\mathrm{e}^{x+y},t)$. The function $f(y)$ is the probability density of a jump of size $y = \log(\eta)$.

It is also common to make the same exponential change of variables $S = \mathrm{e}^x$ in the differential part of the equation:

$$\frac{\partial V}{\partial t} = \frac{\sigma^2}{2} V_{xx} + \left(r - \frac{\sigma^2}{2} - \lambda\kappa \right) \frac{\partial V}{\partial x} - rV + \lambda \int_{-\infty}^{\infty} f(y) V(x+y,t) \mathrm{d}y. \tag{2.3}$$

(This change of variables was used, for example, by Andersen and Andreasen and by Cont.)

Implicit methods such as Crank Nicolson with uniform spacing Δx in space and Δt in time are frequently used to discretize the differential part of the equation:

$$\frac{V^{n+1} - V^n}{\Delta t} = L\left(\frac{V^{n+1} + V^n}{2} \right) + \lambda I(V(x,t)) \tag{2.4}$$

where $V^n = (V_1^n, V_2^n, \ldots, V_m^n)$ is the solution at the nth time step and L is the term involving derivatives with respect to x:

$$L = \frac{\sigma^2}{2} \frac{\partial^2}{\partial x^2} + (r - \lambda\kappa) \frac{\partial}{\partial x} + r.$$

The integrals $I(V(x,t))$ must be evaluated at the differential discretization points $\{x_i\}$. The quadrature points $\{y_i\}$ may be the same points, (Andersen and Andreasen), or else, especially if one wants to use a nonuniform grid for discretizing the differential terms, they may be different [4].

In any case, the integrals are normally approximated by a sum:

$$I(V^n(x_i)) = \sum V_{i+j}^n f_j. \tag{2.5}$$

If one evaluates the integral at the previous time step:

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \frac{1}{2}(L(V_i^{n+1} + V_i^n) + \lambda I(V_i^n))$$

one obtains a tridiagonal linear system of equations at each time step, although the truncation error of the scheme is then $O(\Delta t)$ instead of $O(\Delta t^2)$.

To retain second-order accuracy in time one has to use a fully implicit scheme:

$$\frac{V_i^{n+1} - V_i^n}{\Delta t} = \frac{1}{2} L(V_i^{n+1} + V_i^n) + \lambda \frac{I(V_i^{n+1}) + I(V_i^n)}{2}.$$

This easiest way to solve this equation is by Picard iteration:

$$\frac{V_i^{[n+1,k+1]} - V_i^n}{\Delta t} - \frac{1}{2} L(V_i^{[n+1,k+1]} + V_i^n) = \lambda \frac{I(V_i^{[n+1,k]}) + I(V_i^n)}{2}. \tag{2.6}$$

d'Halluin et al. showed that this iteration is globally convergent when combined with a Crank Nicolson discretization of (2.1). More specifically, they showed that for r and λ nonnegative the iteration will converge if
(1) The off-diagonal terms in the matrix equation arising from the discretization of the differential terms in (2.1) are positive.
(2) For any i, the sum of the weights in (2.5) is less than 1:

$$\sum f_j \leq 1. \tag{2.7}$$

(3) The weights are all nonnegative:

$$g_j \geq 0. \tag{2.8}$$

d'Halluin et al. noted that when pricing single factor options using typical grids and parameter values the first condition is usually satisfied when central differences are used to approximate the first derivatives in (2.1). We also found this was the case in all our calculations. However, as d'Halluin et al. remarked, when this is not the case, one can enforce the condition by using instead upwind differences to approximate the first derivatives. It was also noted in this paper that since upwind differences are normally only used at a few points, the overall rate of convergence is not affected.

As for the other two conditions, if one approximates the integral (2.2) by a standard quadrature method, then conditions (2.7) and (2.8) are easily satisfied. It is not immediately apparent whether they are satisfied when differential methods are used. We discuss these conditions in Section 4.

In contrast, Andersen and Andreasen used an ADI method to solve (2.4). More specifically, they wrote (2.4) as

$$\frac{\partial V}{\partial t} - D_h V = +\lambda f * V$$

where D_h is the discrete approximation to the operator L and where * denotes convolution.

They then solved this equation using the operator splitting:

$$\left(\frac{2}{\Delta t} - D_h\right) V^{n+\frac{1}{2}} = \left(\frac{2}{\Delta t} + \lambda f*\right) V^n$$
$$\left(\frac{2}{\Delta t} - \lambda f*\right) V^{n+1} = \left(\frac{2}{\Delta t} + D_h\right) V^{n+\frac{1}{2}}.$$

In their calculations they let D_h be the usual centered difference approximation to L. This method is second-order accurate in space and time.

Whichever discretization of the differential part of the equation and solution strategy one is using one needs to evaluate the integral term. Since it is a correlation integral, one way to do this is by using the Fast Fourier Transform. That is, in order to evaluate (2.2) one can compute the Fourier transform $\hat{V}$ of $\{V_i^n\}$, from $R = \hat{V}\hat{f}$, and then compute the inverse Fourier transform of R. The cost is essentially the cost of two FFTs.

In the next two sections we present alternate, more rapid ways of evaluating the integrals and of solving the PIDE for exponential-type and Merton-type densities.

3. Operator splitting and the Kou model

In Kou's model [10] the density is a double exponential:

$$f(x) = p\eta_1 \mathrm{e}^{-\eta_1 x} H(x) + q\eta_2 \mathrm{e}^{\eta_2} H(-x)$$

where $\eta_1 > 1$, $\eta_2 > 0$, $p > 0$, $q = p - 1$, and H is the Heaviside function. In this model the expected value of the jump size is

$$\kappa = \frac{p\eta_1}{\eta_1 - 1} + \frac{q\eta_2}{\eta_2 + 1} - 1.$$

Kou has shown that in addition to having the leptokurtic feature and volatility smiles, the model has analytic solutions in terms of the Laplace transform and Hh function for a variety of option pricing problems, including barriers and perpetual Americans.

As noted in the previous section, because of their stability properties implicit methods are often used for solving the pricing PIDE, and since the equations contain the convolution of the jump density with the (unknown) option price, the resulting equations are commonly solved iteratively. In this section we show how to solve the equations more rapidly.

We first show how to efficiently evaluate the jump integrals in the Kou model. In the Kou model the pricing PIDE contains integrals of the form

$$\int_0^\infty V(x+y,t)\mathrm{e}^{-\eta_1 x}\mathrm{d}x \quad \text{and} \quad \int_{-\infty}^0 V(x+y,t)\mathrm{e}^{\eta_2 x}\mathrm{d}x.$$

Since the integrals are correlation integrals they are frequently computed using the fast Fourier transform. Fourier transform methods require enlargement of the computational domain, and for efficiency normally require the number of mesh points to be a power of 2. More serious difficulties also arise because of the discontinuity of the density (the Gibb's phenomenon). This means that the convergence rate will be very slow.

In order to evaluate the integrals more rapidly we first note that after changes of variables these and similar integrals can be expressed as products of exponentials and integrals of only one variable.

Specifically, under the additional change of variables $s = x + y$ the first integral becomes

$$J_1 V(y,t) = \mathrm{e}^{\eta_1 y}\int_y^\infty V(s,t)\mathrm{e}^{-\eta_1 s}\mathrm{d}s = \mathrm{e}^{\eta_1 y} I_1(y), \tag{3.1}$$

and the second becomes

$$J_2 V(y,t) = \mathrm{e}^{-\eta_2 y}\int_{-\infty}^y V(s,t)\mathrm{e}^{\eta_2 s}\mathrm{d}s = \mathrm{e}^{-\eta_2 y} I_2(y). \tag{3.2}$$

To compute $I_1(y)$ we first approximate

$$I_0 = \int_0^\infty V(s,t)\mathrm{e}^{-\eta_1 s}\mathrm{d}s$$

using a uniformly spaced grid $(y_0, y_1, \ldots, y_m)$:

$$I_0 \cong \sum_0^m w_i V(y_i,t)\mathrm{e}^{-\eta_1 y_i}.$$

The point y_m is chosen so that $\mathrm{e}^{-\eta y_m} V(y_m)$ is sufficiently small. Then, to evaluate $I_1(y)$ we note that

$$I_1(y) = I_0 - \int_0^y V(s,t)\mathrm{e}^{-\eta_1 s}\mathrm{d}s. \tag{3.3}$$

We compute $I_1(y)$ inductively:

$$I_1(y_i) \cong I_1(y_{i-1}) + w_i V(y_i)\mathrm{e}^{-\eta_1 y_i}.$$

The integral $I_2(y)$ is computed in the same manner, so the entire calculation is linear in the number of evaluation points.

We note that essentially the same method can be used to evaluate the correlation integral if the jump density $f(x)$ is the product of any polynomial and an exponential since the same linear change of variables makes the kernels separable.

We now present a rapid new method for solving the pricing PIDE in the Kou model.

We recall that by using the operator splitting proposed by Andersen and Andreasen for pricing option in the Merton model at every time step we obtain two (consecutive) equations to solve, each of which can be solved directly, i.e. no iteration is needed.

More precisely, the first equation has the form

$$\left(\frac{2}{\Delta t} I - D_h\right) V^{n+\frac{1}{2}} = \left(\frac{2}{\Delta t} I + \lambda f *\right) V^n = \frac{2}{\Delta t} V^n + \lambda \int f * V^n. \tag{3.4}$$

In contrast to Andersen and Andreasen who used Fourier methods (for pricing in the Merton model), we use the above method to evaluate the integral on the right-hand side. In their method and ours, when proper boundary conditions are provided and the integral has been evaluated, the solution of (3.4) reduces to the solution of a tridiagonal linear system of equations.

In the Kou model the second equation has the form

$$aV^{n+1}(y) + c_2 \int_{-\infty}^{y} e^{\eta_2(s-y)} V^{n+1}(s)\mathrm{d}s + c_1 \int_{y}^{\infty} e^{-\eta_1(s-y)} V^{n+1}(s)\mathrm{d}s$$
$$= aV^{n+1} + c_2 J_2(y) + c_1 J_1(y) = g(y) \tag{3.5}$$

where $a = \frac{2}{\Delta t}$, $c_1 = \eta p\lambda$, $c_2 = \eta q\lambda$ and $g(y) = (\frac{2}{\Delta t} + D_h)V^{n+\frac{1}{2}}$.

Andersen and Andreasen also used Fourier methods to solve this equation when pricing in the Merton model. For pricing options in the Kou model we instead reduce the solution of this equation to the solution of a second-order boundary value problem (ODE). We first note that

$$\frac{\mathrm{d}J_1(V^{n+1}(y))}{\mathrm{d}y} = \eta_1 \int_{y}^{\infty} e^{-\eta_1(s-y)} V^{n+1}(s)\mathrm{d}s - V^{n+1}(y) = \eta_1 J_1(V^{n+1}(y)) - V^{n+1}(y)$$

and

$$\frac{\mathrm{d}J_2(V^{n+1}(y))}{\mathrm{d}y} = -\eta_2 J_2(y) + V^{n+1}(y).$$

By the above equations

$$L_1 J_1 V^{n+1}(y) = -V^{n+1}(y), \qquad L_1 J_2 V^{n+1}(y) = -(\eta_1 + \eta_2)V^{n+1}(y),$$
$$L_2 J_1 V^{n+1}(y) = (\eta_1 + \eta_2)V^{n+1}(y), \quad \text{and} \quad L_2 J_2 V^{n+1}(y) = V^{n+1}(y)$$

where

$$L_1 w(y) = \frac{\mathrm{d}w(y)}{\mathrm{d}y} - \eta_1 w(y)$$

and

$$L_2 w(y) = \frac{\mathrm{d}w(y)}{\mathrm{d}y} + \eta_2 w(y).$$

Therefore, by first applying the operator L_1 to (3.5) we obtain

$$aL_1 V^{n+1}(y) - c_2(\eta_2 + \eta_1)J_2(y) + (c_2 - c_1)V^{n+1}(y) = L_1 g(y), \tag{3.6}$$

and by next applying the operator L_2 to (3.6) we obtain

$$aL_2 L_1 V^{n+1}(y) - c_2(\eta_2 + \eta_1)V^{n+1}(y) + (c_2 - c_1)L_2 V^{n+1}(y) = L_2 L_1 g(y). \tag{3.7}$$

Since

$$L_2 L_1 f(y) = \frac{\mathrm{d}^2 f(y)}{\mathrm{d}y^2} + (\eta_2 - \eta_1)\frac{\mathrm{d}f(y)}{\mathrm{d}y} - \eta_1\eta_2 f(y),$$

Eq. (3.7) has the form

$$\frac{\mathrm{d}^2 V^{n+1}(y)}{\mathrm{d}y^2} + b\frac{\mathrm{d}V^{n+1}(y)}{\mathrm{d}y} + aV^{n+1}y = \tilde{g}(y) \tag{3.8}$$

where $b = (\eta_2 - \eta_1) + (c_2 - c_1)$ and $a = (\eta_1 c_2 + \eta_2 c_1) - \eta_1\eta_2$ and $\tilde{g} = L_1 L_2 g(V^{n+1/2})$.

We have thus obtained a second-order linear boundary value problem with constant coefficients which we can solve for $V^{n+1}(y)$. (We also note Eq. (3.8) has an analytic solution containing integrals involving $\tilde{g}$.)

In order to solve (3.4) and (3.8) we need to supply boundary conditions. As d'Halluin et al. [7] have noted we can use the same boundary conditions for solving the pricing PIDE as when no jumps are present. In particular, for a vanilla call we set $V^{n+1}(0) = 0$, and we require $V^{n+1}(S)$ to be linear for S sufficiently large. The boundary conditions for $V^{n+1/2}$ are obtained in the standard manner for a Peaceman Rachford ADI algorithm, that is we add (3.4) and (3.5), and use the above conditions for V^n and V^{n+1}. If D_h is the usual second-order difference equation, with these boundary conditions the solutions of (3.4) and (3.8) reduce to the solutions of tridiagonal linear systems of equations.

We now consider the numerical effects of solving (3.8) instead of (3.5). It turns out that we do not normally lose accuracy at points where the gradient of the solution is large. However, in order to maintain accuracy we need to have accurate estimates for more derivatives of the solution at the ends of the computational domain. We have found that this may require us to extend the domain and use more mesh points, which, of course, increases the cost of the calculation. However, not having to enlarge the region to avoid wrap around and not having to be concerned with the decreased convergence rate due to the Gibb's phenomenon result in much larger savings.

In order to understand why there usually is not a significant loss of accuracy in solving what appears to be a higher order differential equation we first note that if we let A_h denote the discrete integral operator, then Andersen and Andreasen splitting can be written as

$$\left(\frac{\Delta t}{2}I - D_h\right) V^{n+\frac{1}{2}} = \left(\frac{\Delta t}{2}I + A_h\right) V^n \tag{3.9}$$

$$\left(\frac{\Delta t}{2}I - A_h\right) V^{n+1} = \left(\frac{\Delta t}{2}I + D_h\right) V^{n+\frac{1}{2}}. \tag{3.10}$$

If we let L_h denote the centered difference approximation to the second-order differential operator $L_1 L_2$, our computational method can be viewed as applying the *discrete* operator L_h to both sides of (3.10):

$$L_h\left(\frac{\Delta t}{2}I - A_h\right) V^{n+1} = L_h\left(\frac{\Delta t}{2}I + D_h\right) V^{n+\frac{1}{2}}, \tag{3.11}$$

and using the approximation

$$L_h A_h(V^{n+1}) = L_h\left(c_1 J_1(V^{n+1}) + c_2 J_2(V^{n+1})\right) = -c_2(\eta_2 + \eta_1)V^{n+1} + (c_2 - c_1)L_{2h}V^{n+1}. \tag{3.12}$$

(Here L_{2h} is the centered difference approximation to L_2.)

We see that if correct boundary values are provided for Eq. (3.11) it will have the same solution as (3.10), up to the difference in truncation error that occurs due to replacing $L_h A_h V$ using (3.12). Since A_h is an integral operator, and therefore smoothing, one does not expect this difference in truncation error to be large.

We also note that if there is no integral in the original equation, and if D_h commutes with L_h, (for example if volatility is not a function of S,), then whenever boundary values of V and their derivatives are known precisely the splitting we obtain is equivalent to the Crank Nicolson discretization. That is, if we use the exact values of $L_h V^n$ at the ends of the computational region, the solution of the system

$$\left(\frac{\Delta t}{2}I - D_h\right) V^{n+\frac{1}{2}} = \frac{\Delta t}{2}V^n$$

$$L_h\frac{\Delta t}{2}V^{n+1} = L_h\left(\frac{\Delta t}{2}I + D_h\right) V^{n+\frac{1}{2}}$$

is the same as the solution of the Crank Nicolson equation:

$$\left(\frac{\Delta t}{2}I - D_h\right) V^{n+1} = \left(\frac{\Delta t}{2}I + D_h\right) V^n.$$

More generally, if we apply a difference operator M_h to any mesh function $\{G_i\}$, and then invert M_h, the result will be $\{G_i\}$ as long as we have computed $M_h G_i$ properly at the ends of the computational region, even if the values of $\{G_i\}$ are the values of a discontinuous function.

At each time step the second equation of the operator splitting of the Crank Nicolson discretization of the pricing PIDE also reduces to a boundary value problem when the density has components of the form $f(x) = x^n e^{\eta x}$.

For example, suppose that

$$f(x) = c_2 x e^{\eta_2 x} H(-x) + c_1 x e^{-\eta_1 x} H(x).$$

In this case the second equation in the splitting has the form

$$aV^{n+1} + c_2 \int_{-\infty}^{y} (s-y) e^{\eta_2 (s-y)} V^{n+1}(s) ds + c_1 \int_{y}^{\infty} (s-y) e^{-\eta_1 (s-y)} V^{n+1}(s) ds = g. \tag{3.13}$$

We let

$$J_3 V^{n+1}(y) = \int_{y}^{\infty} (s-y) e^{-\eta_1 (s-y)} V^{n+1}(s) ds,$$

and

$$J_4 V^{n+1}(y) = \int_{-\infty}^{y} (s-y) e^{\eta_2 (s-y)} V^{n+1}(s) ds,$$

so (3.13) has the form:

$$aV^{n+1} + c_2 J_3 V^{n+1}(y) + c_1 J_4 V^{n+1}(y) = g. \tag{3.14}$$

Since

$$\frac{dJ_3(V^{n+1})}{dy} = \int_{y}^{\infty} \left(\eta_1 (s-y) e^{-\eta_1 (s-y)} - e^{-\eta_1 (s-y)} \right) V^{n+1}(s) ds,$$

$$\frac{d^2 J_3}{dy^2} = \int_{y}^{\infty} \left(\eta_1^2 (s-y) e^{-\eta_1 (s-y)} - 2\eta_1 e^{-\eta_1 (s-y)} \right) V^{n+1}(s) ds + V^{n+1}(y),$$

$$\frac{dJ_4 V^{n+1}}{dy} = \int_{-\infty}^{y} \left(-\eta_2 (s-y) e^{\eta_2 (s-y)} - e^{\eta_2 (s-y)} \right) V^{n+1}(s) ds \quad \text{and}$$

$$\frac{d^2 J_4 V^{n+1}}{dy^2} = \int_{-\infty}^{y} \left(\eta_2^2 (s-y) e^{\eta_2 (s-y)} + 2\eta_2 e^{\eta_2 (s-y)} \right) V^{n+1}(s) ds - V^{n+1}(y)$$

it follows that

$$L_3(J_3 V^{n+1}(y)) = V^{n+1}(y)$$
$$L_4(J_4 V^{n+1}(y)) = -V^{n+1}(y)$$

where

$$L_3 w = w'' - 2\eta_1 w' + \eta_1^2 w$$
$$L_4 w = w'' + 2\eta_2 w' + \eta_2^2 w.$$

Therefore, by applying $L_3 L_4$ to (3.13) we obtain the boundary value problem

$$L_3 L_4 V^{n+1} + c_2 L_4 V^{n+1} - c_1 L_3 V^{n+1} = L_3 L_4 g(V^{n+1/2}).$$

As for boundary conditions, they are the same as for the Black–Scholes differential equation [7]. Thus, for example, for pricing a vanilla call, in addition to the usual conditions we can require any second or higher order derivatives of the option price to vanish if the asset price is sufficiently small or large. Similarly, we can obtain linear constant coefficient differential equations at each time step when the density has components of the form $x^n \sin(bx) e^{ax}$ or $x^n \cos(bx) e^{ax}$. We also note that many density functions that are not of that form can be uniformly approximated by linear combinations of such functions on different subintervals.

4. Merton model and extensions

In Merton's model the probability density function of the jumps is

$$g(\eta) = \frac{e^{\left(-\frac{(\log(\eta)-\mu)^2}{2\gamma^2}\right)}}{\sqrt{2\pi}\gamma\eta}.$$

The expected relative change in the stock price is $\kappa = E[\eta - 1] = e^{\mu+\gamma^2/2} - 1$.

For μ, σ and γ constant the price of a European nonpath-dependent option is an infinite sum:

$$\sum_{n=0}^{\infty} \frac{1}{n!} e^{\lambda'(T-t)} (\lambda'(T-t))^n V_{BS}(S, t, \sigma_n, r_n),$$

where

$$\lambda' = \lambda(1+\kappa), \qquad \sigma_n^2 = \sigma^2 + \frac{n\gamma^2}{T-t}, \qquad r_n = r - \lambda k + \frac{n\log(1+\kappa)}{T-t},$$

and V_{BS} is the Black–Scholes formula for the option value in the absence of jumps. Usually only the first few terms in the sum are needed to obtain 6 digits of accuracy in the option price. However, when the Black–Scholes price is not known, or the option is not European, numerical methods are needed to price the option. These methods generally require computation of the correlation integral. Fourier methods can be used, but are expensive. We now show how to decrease the cost of computing the correlation integrals and avoid other numerical difficulties.

After making the exponential change of variables the integral term in the pricing equation can be written as

$$I(V(x,t)) = \int_{-\infty}^{\infty} V(x+y,t) \frac{e^{-\frac{(y-\mu)^2}{2\gamma^2}}}{\sqrt{2\pi}\gamma} dy.$$

By further letting $s = x + y$ we see that

$$I(V(x,t)) = \frac{1}{2\pi\gamma} \int_{-\infty}^{\infty} V(s,t) e^{-\frac{(s-x-\mu)^2}{2\gamma^2}} ds. \tag{4.1}$$

The kernel in (4.1) is a translation of the fundamental solution of the heat equation

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} \tag{4.2}$$

on the line $-\infty \le x \le \infty$, where τ is a nonphysical variable corresponding to half of the variance of the process, $\frac{\gamma^2}{2}$. (That is, we view τ as a continuous variable, $0 \le \tau \le \gamma^2/2$.) It follows that we can evaluate the integral (4.1) at any fixed time t by solving the heat equation with initial data $V(x,t)$ at $\tau = 0$. We evaluate the solution of (4.2) at $\tau = \frac{\gamma^2}{2}$, and then translate it by the amount μ.

In our numerical tests we evaluated the integral (4.1) by solving the heat equation using a discretization that is fourth-order accurate in space and second-order accurate in τ, and then using second-order Richardson extrapolation with respect to τ. Thus, the method was fourth-order accurate in both x and τ.

The discretization combines the usual centered difference in time and a weighted average of the centered differences in space at the current and the previous time steps:

$$\frac{u_i^{n+1} - u_i^n}{\Delta\tau} = \theta\delta_x^2 u_i^{n+1} + (1-\theta)\delta_x^2 u_i^n \tag{4.3}$$

where

$$\delta_x^2 u_i^n = \frac{u_{i-1}^n - 2u_i^n + u_{i+1}^n}{\Delta x^2},$$

and the weight $\theta = \frac{1}{2} - \frac{(\Delta x)^2}{12\Delta\tau}$ depends on the relative mesh spacing (Richtmeyer and Morton).

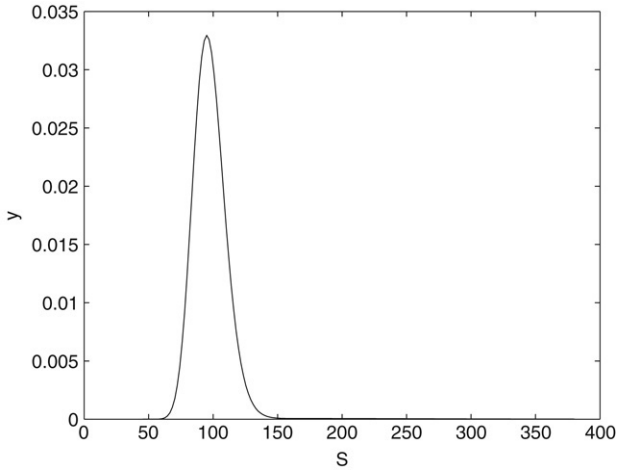

Fig. 1. Second derivative with respect to S of convolution integral.

If we take N steps in τ, we need to provide boundary conditions for u at $\tau_j = \frac{j\gamma^2}{2N}$. In order to price call options we used homogeneous Dirichlet boundary conditions at the left end point, and required the solution to have zero curvature with respect to S at the right end point, that is we set $\frac{\partial^2 u}{\partial S^2} = 0$.

It is important that for these boundary conditions to be accurate the computational region need not be large. In Fig. 1 we have plotted $y = \frac{\partial^2 I(V(S,t))}{\partial S^2}$ for the call option with $\gamma = .45$, $\sigma = .15, t = .25, r = .05$, strike $E = 100$, $\mu = -.30$ and $\lambda = .1$. Thus we see that $\frac{\partial^2 I(V(S,t))}{\partial S^2}$ is close to 0 for S relatively small.

At each step in τ the discretization (4.3) leads to a tridiagonal linear system of equations whose solution only requires 3 operations per point. Since we are using extrapolation with respect to τ the total cost of solving Eq. (4.2) is $4.5Nm$ where m is the number of mesh points in the x direction. (3 operations per point to solve (4.3) plus 1.5 operations per point for extrapolation) This is in contrast to two Fourier transforms, i.e. a cost of approximately $10m \log m$ for computing the integral using Fourier methods. However, it is important to note that when using differential methods we need not enlarge the computational region to avoid problems with wrap around. That is, the number of mesh points in x is usually much smaller in our method of calculation. Typically m must be chosen to be 1000 or larger in order to obtain 4–5 digits of accuracy when solving the pricing PIDE when Fourier methods are used to evaluate integrals, so a total of at least 100,000 operations are needed. In contrast, we found that we could obtain similar accuracy in evaluating the integrals with $m = 200$ and $N = 10$, that is 9000 operations. We note that we do not expect the need to take N larger than 10, since our method is fourth-order accurate in τ, and γ is typically less than .5. Therefore $\mathrm{d}\tau = \frac{\gamma}{2N} \le .0125$, and so $(\mathrm{d}\tau)^4 \le 10^{-7}$.

In addition to the expense of Fast Fourier transform methods there is another problem with computing the integral (4.1) by evaluating the sum

$$I(V_i^n) = \sum_{j=-\infty}^{\infty} V_j^n f_{i-j} \tag{4.4}$$

where

$$f_j = \frac{1}{\gamma\sqrt{2\pi}} \int_{x_i - y_{j-1/2}}^{x_i - y_{j+1/2}} \mathrm{e}^{-\frac{(y-\mu)^2}{2\gamma^2}} \mathrm{d}y.$$

In order to approximate the integral the sum must be truncated. However, if γ is large, f_{i-j} is not small unless $x_i - y_j$ is very large. This means that in order to obtain high accuracy, even near the center of the computational region, the region of integration must be very large. This is a common problem when integral methods are used to solve the heat equation [15]. On the other hand, if γ is small, f_{i-j} is large for i close to j, so any errors made in evaluating V_i will be magnified. In contrast, the values of the integral $I(V(x,t,\gamma))$ are smooth and easily computed by differential methods.

Table 1
Prices of European Call in Kou model

m	nt	S	Computed values	Error	R
120	20	90	0.674993	0.002316	–
240	40	90	0.673288	0.000611	1.92
480	80	90	0.672843	0.000166	1.84
960	160	90	0.672720	0.000043	1.94
120	20	100	3.986678	0.013199	–
240	40	100	3.977150	0.003671	1.82
480	80	100	3.974437	0.000958	1.92
960	160	100	3.973722	0.000243	1.97
120	20	110	11.797584	0.003001	–
240	40	110	11.795383	0.000800	1.88
480	80	110	11.794802	0.000219	1.83
960	160	110	11.794641	0.000058	1.90

We also note that if we use this differential method of approximating $I(V_i)$ conditions (2.7) and (2.8) for the convergence of the Picard iteration (2.6) can be satisfied. Condition (2.7) follows if the scheme used to solve the heat is monotone, i.e. $u_i^{n+1} \geq 0$ for all i if $u_i^n \geq 0$ for all i. If one is solving an initial value problem for the heat equation with a uniform mesh it is known (see [6]) that the scheme (4.3) is monotone if

$$4\frac{\Delta\tau}{(\Delta x)^2} \leq \frac{2-\theta}{(1-\theta)^2}.$$

This equation is satisfied if $18\alpha^2 - 21\alpha \leq 1$ where $\alpha = \frac{\Delta\tau}{(\Delta x)^2}$. (For bounded regions the restriction is slightly stronger.) Thus, for a given mesh width in x, if the time steps are sufficiently small the scheme will be monotone. Since we normally used large mesh width in x, in the majority of our numerical experiments, this was the case. However, even when the condition was not satisfied, the Picard iteration (2.6) was still convergent.

In order to show that condition (2.8) is satisfied by our method, we let $E_{k,i}^N$ be the solution at x_k of the discrete heat equation (2.5) with initial data $u_i^0 = 1, u_j^0 = 0$ for $j \neq i$ at the N th time step. Since $\sum_{i=1}^m E_{k,i}$ is the solution at x_k of the heat equation with initial values 1 at all mesh points, it is less than or equal to 1. (This follows by direct calculation or the discrete maximum principle.) By linearity $I(V_k) = \sum_i V_i E_{k,i}$, so we see (2.8) is satisfied.

Since t is a constant in the integral (4.1) we can also use the same method of evaluating correlation integrals if the mean μ and variance γ of the process are time dependent.

We can also evaluate jump integrals when the density function of the process is the product of a Gaussian and a polynomial or the product of a Gaussian and an exponential. That is because the correlation integrals are combinations of solutions of heat equations or their translations.

5. Numerical results

In this section we report on results of numerical tests we performed solving the pricing PIDEs in jump-diffusion models and evaluating the jump integrals.

In Table 1 we present results of using our pricing method to value a European vanilla call option in the Kou model. We used Simpson's rule as the quadrature formula to evaluate the integrals (3.1) and (3.2), with input parameters $\sigma = .15, r = .05, \lambda = .1, T = .25, E = 100, \eta_1 = 3.0465, \eta_2 = 3.0465$, and $p = .3445$. The numbers in the first column, m are the number of steps in x, the numbers nt in the second column are the number of time steps, the numbers in the third column are the asset prices S, the numbers in the fourth column are the computed values of the option price, the numbers in the next column are the errors, and the numbers in the last column are the convergence rate. We note that the exact value of the option at $S = 90$ is .672677, at $S = 100$ it is 3.973479, and at $S = 110$ it is 11.794583.

We see that the method is essentially second-order accurate.

In the next table, Table 2, we give results of pricing an American put option in the Kou model using our modified splitting method. We enforced the free boundary condition when solving the second equation, i.e. (3.8), using a

Table 2
Prices of American Put in Kou model

m	nt	S	Computed values	Error	R
64	10	90	10.050761	0.045690	–
128	20	90	10.019567	0.014496	1.66
256	40	90	10.009184	0.004113	1.82
512	80	90	10.006218	0.001147	1.84
1024	160	90	10.005379	0.000308	1.90
64	10	100	2.905190	0.097312	–
128	20	100	2.840808	0.032930	1.56
256	40	100	2.797957	0.009921	1.73
512	80	100	2.804946	0.002932	1.76
1024	160	100	2.808767	0.000889	1.72
64	10	110	0.620063	0.058187	–
128	20	110	0.544711	0.017165	1.76
256	40	110	0.566504	0.004628	1.89
512	80	110	0.560539	0.001337	1.79
1024	160	110	0.562257	0.000381	1.81

projected SOR method. The parameters are the same as in the previous table, i.e. $\sigma = .15, r = .05, \lambda = .1, T = .25, E = 100, \eta_1 = 3.0465, \eta_2 = 3.0465$, and $p = .3445$.

In our next set of experiments we tested our method for evaluating the integral (4.1) that is part of the pricing equation in the Merton model. We note that since the analytic solution of the pricing equation is known, we could evaluate the integral analytically. That is, since the price $V(S, t)$ is known for a given set of parameters $\sigma, r, T, E, \lambda, \gamma, t$ and μ, we could evaluate

$$I(V(S,t)) = -\frac{1}{\lambda}\left(-V_t + \frac{\sigma^2 S^2}{2} V_{SS} + (r - \lambda\kappa) S V_S - (r + \lambda)\, V\right).$$

We next tested the accuracy of our method of evaluating the integral by solving the heat equation (4.2) using the discretization (4.3). In Tables 3 and 4 are results of our calculations evaluating the correlation integral $I(V(S, t))$ for the European call option with time to expiration $T = 3$ months, volatility $\sigma = .15$, interest rate $r = .05$, strike price $E = 100$, $\mu = -.30$, and $\lambda = .1$. Results in Table 3 are for $\gamma = .45$, and results in Table 4 are for $\gamma = .25$.

In all cases the number of x mesh points we used for solving (4.2) was 200, and the number of τ steps was 10. The numbers in column 1 are the asset prices, the numbers in the second column are the computed values of the integral, the numbers in the third column are the exact values of the integral, the numbers in the fourth column are the values of the option, the numbers in the fifth column are the errors and the numbers in the last column are the relative errors, that is, the errors divided by the option values.

We see that, as expected, we could compute the integral values to 5 digits of accuracy using only 200 mesh points.

Next we evaluated the integrals $I(V(S, t))$ by direct summation as in Eq. (4.4). We chose the same option parameters as in Tables 3 and 4. That is, results in Table 5 are for $\gamma = .45$, and those in Table 6 are for $\gamma = .25$. We again used 200 mesh points in x.

We see that these results are significantly less accurate than those reported in Tables 3 and 4. However, we note that by expanding the region of integration we could have achieved slightly better results, especially for $\gamma = .45$.

We also solved the entire pricing PIDE for a European call. The results in Table 7 are for the option parameters time to expiration $T = 3$ months, volatility $\sigma = .25$, interest rate $r = .05$, strike price $E = 100$, $\mu = -.90$, $\lambda = .1$ and $\gamma = .25$. We used 15 steps in τ.

For $m = 32, 64$, and 128 we evaluated the integral by solving the heat equation by the fourth-order accurate method (4.3), which was monotone for the values of Δx and $\Delta\tau$ used. We used 15 steps in τ. For $m = 268$ and 512 we evaluated the integral by solving the heat equation using the explicit scheme that is second-order accurate in x and first-order accurate in τ. Here we used 20 steps in τ. However, the scheme still converged when we used the nonmonotone higher order method. (This may have occurred because for a call option most of the truncation error has the same sign.)

Table 3
Evaluation of integral in Merton model by solving differential equation, $\gamma = .45$

S	Computed value	Exact value	Option value	Error	Relative error
89.42	0.6025	0.6024	0.4865	0.59E−04	0.12E−03
91.10	0.6502	0.6502	0.7221	0.21E−04	0.29E−04
92.81	0.7009	0.7009	1.0659	0.30E−04	0.28E−04
94.56	0.7548	0.7549	1.5463	0.91E−04	0.59E−04
96.34	0.8119	0.8120	2.1898	0.16E−03	0.71E−04
98.15	0.8724	0.8727	3.0165	0.22E−03	0.73E−04
100.00	0.9365	0.9368	4.0379	0.27E−03	0.67E−04
101.88	1.0043	1.0046	5.2547	0.29E−03	0.56E−04
103.80	1.0759	1.0762	6.6578	0.28E−03	0.42E−04
105.75	1.1515	1.1518	8.2296	0.21E−03	0.25E−04
107.74	1.2313	1.2314	9.9476	0.79E−04	0.80E−05
109.77	1.3153	1.3152	11.7881	0.11E−03	0.90E−05
111.83	1.4036	1.4033	13.7287	0.33E−03	0.24E−04

Table 4
Evaluation of integral in Merton model by solving differential equation, $\gamma = .25$

S	Computed value	Exact value	Option value	Error	Relative error
88.83	0.0751	0.0751	0.3097	0.26E−06	0.85E−06
90.60	0.0894	0.0894	0.5268	0.30E−06	0.58E−06
92.41	0.1060	0.1060	0.8578	0.19E−05	0.22E−05
94.25	0.1251	0.1251	1.3367	0.44E−05	0.33E−05
96.13	0.1471	0.1471	1.9952	0.71E−05	0.35E−05
98.05	0.1722	0.1722	2.8580	0.89E−05	0.31E−05
100.00	0.2008	0.2008	3.9388	0.87E−05	0.22E−05
101.99	0.2332	0.2332	5.2381	0.53E−05	0.10E−05
104.03	0.2697	0.2697	6.7437	0.14E−05	0.21E−06
106.10	0.3107	0.3107	8.4338	0.10E−04	0.12E−05
108.21	0.3566	0.3566	10.2813	0.18E−04	0.17E−05
110.37	0.4076	0.4077	12.2580	0.21E−04	0.17E−05

Table 5
Evaluation of integral in Merton model by quadrature, $\gamma = .45$

S	Computed value	Exact value	Option value	Error	Relative error
89.42	0.5956	0.6024	0.4865	0.68E−02	0.14E−01
91.10	0.6422	0.6502	0.7221	0.80E−02	0.11E−01
92.81	0.6916	0.7009	1.0659	0.93E−02	0.87E−02
94.56	0.7440	0.7549	1.5463	0.11E−01	0.70E−02
96.34	0.7994	0.8120	2.1898	0.13E−01	0.58E−02
98.15	0.8580	0.8727	3.0165	0.15E−01	0.49E−02
100.00	0.9198	0.9368	4.0379	0.17E−01	0.42E−02
101.88	0.9850	1.0046	5.2547	0.20E−01	0.37E−02
103.80	1.0535	1.0762	6.6578	0.23E−01	0.34E−02
105.75	1.1256	1.1518	8.2296	0.26E−01	0.32E−02
107.74	1.2012	1.2314	9.9476	0.30E−01	0.30E−02
109.77	1.2805	1.3152	11.7881	0.35E−01	0.29E−02
111.83	1.3635	1.4033	13.7287	0.40E−01	0.29E−02

Finally we note that because the Gaussian density decays more rapidly than the Kou density we could use fewer mesh points in x to achieve the same level of accuracy when pricing these options in comparison to our calculations in the Kou model. However, the number of operations per point was larger, so the entire calculation was still still more expensive.

Table 6
Evaluation of integral in Merton model by quadrature, $\gamma = .25$

S	Computed value	Exact value	Option value	Error	Relative error
88.83	0.0751	0.0751	0.3097	0.84E−04	0.27E−03
90.60	0.0895	0.0894	0.5268	0.95E−04	0.18E−03
92.41	0.1061	0.1060	0.8578	0.11E−03	0.12E−03
94.25	0.1252	0.1251	1.3367	0.12E−03	0.89E−04
96.13	0.1472	0.1471	1.9952	0.13E−03	0.66E−04
98.05	0.1724	0.1722	2.8580	0.15E−03	0.51E−04
100.00	0.2010	0.2008	3.9388	0.16E−03	0.40E−04
101.99	0.2334	0.2332	5.2381	0.17E−03	0.33E−04
104.03	0.2699	0.2697	6.7437	0.19E−03	0.28E−04
106.10	0.3109	0.3107	8.4338	0.20E−03	0.24E−04
108.21	0.3568	0.3566	10.2813	0.22E−03	0.21E−04
110.37	0.4079	0.4077	12.2580	0.23E−03	0.19E−04

Table 7
Pricing European call in Merton model, $\gamma = .25$

m	nt	S	Computed values	Error	R
32	10	90	1.9165981	0.0103529	–
64	20	90	1.9239608	0.0029902	1.79
128	40	90	1.9261074	0.0008436	1.83
256	80	90	1.9271940	0.0002430	1.80
512	160	90	1.9268646	0.0000864	1.49
32	10	100	6.4191850	0.0143770	–
64	20	100	6.4296335	0.0039285	1.87
128	40	100	6.4345348	0.0009728	2.01
256	80	100	6.4332149	0.0003471	1.49
512	160	100	6.4336609	0.0000989	1.81
32	10	110	13.8320110	0.0086590	–
64	20	110	13.8385481	0.0021219	2.03
128	40	110	13.8400615	0.0006085	1.80
256	80	110	13.8405196	0.0001504	2.02
512	160	110	13.8406295	0.0000405	1.89

6. Conclusions

When the underlying asset follows a jump-diffusion process the price of an option on that asset satisfies a partial integral differential equation. The primary difficulty encountered when trying to solve such equations numerically is the presence of the nonlocal, integral terms. In this paper we have shown how to reduce the cost of evaluating such integrals when the density functions of the jump processes are products of polynomials, exponentials and Gaussians. We have also shown how to use the operator splitting method proposed by Andersen and Andreasen to reduce the solution of the pricing equation in the Kou and similar models to a sequence of ordinary differential equations. In future work we plan to extend these methods to pricing options whose values depend on several assets.

References

[1] K. Amin, Jump diffusion option valuation in discrete time, Journal of Finance 48 (1993) 1833–1863.
[2] L. Andersen, J. Andreasen, Jump diffusion processes: Volatility smile fitting and numerical methods for option pricing, Review of Derivatives Research 4 (2000) 231–262.
[3] P. Carr, D. Madan, H. German, M. Yor, From local levy to local volatility models, Quantitative Finance 4 (2004) 581–588.
[4] R. Cont, E. Voltchkova, A finite difference scheme for option pricing in jump diffusion and exponential levy models, Internal Report 513, CMAP, Ecole Polytechnique, 2003.
[5] R. Cont, P. Tankov, Non-parametric calibration of jump-diffusion option pricing models, Journal of Computational Finance 7 (3) 1–49.
[6] L. Dautray, J. Lions, Mathematical Analysis and Numerical Methods for Science and Technology, vol. 6, Springer Verlag, Berlin, 1990.

[7] Y. d'Halluin, P.A. Forsyth, K.R. Vetzal, Numerical methods for contingent claims under jump diffusion processes, IMA Journal of Numerical Analysis 25 (2005) 87–112.
[8] Y. d'Halluin, P.A. Forsyth, G. Labahn, A penalty method for American options with jump diffusion processes, Numerische Mathematik 97 (2004) 321–352.
[9] Y. d'Halluin, P.A. Forsyth, G. Labahn, A semi-Lagrangian approach for American Asian options under jump diffusion, SIAM Journal Scientific Computing 27 (2005) 315–345.
[10] S. Kou, A jump diffusion model for option pricing, Management Science 48 (2002) 1086–1101.
[11] P. Madan, P. Carr, E. Chang, The variance gamma process and option pricing, European Financial Review 1 (1998) 75–105.
[12] A. Matache, T. von Petersdorff, C. Schwab, Fast deterministic pricing of options on Levy driven assets, RiskLab Research Report, ETH, Zurich, 2002.
[13] R.C. Merton, Option pricing when underlying stock returns are discontinuous, Journal of Financial Economics 3 (1976) 125–144.
[14] R. Richtmeyer, K. Morton, Difference Methods for Initial Value Problems, Wiley, New York, 1967.
[15] J. Strickwerda, Finite Difference Schemes and Differential Equations, Wadsworth and Brooks, Pacific Grove, 1989.
[16] D. Tavella, C. Randall, Pricing Financial Instruments: The Finite Difference Method, John Wiley and Sons, Inc., 2000.
[17] P. Wilmott, Derivatives, John Wiley and Sons, 1998.
[18] X. Zhang, Optiones Americaines et Modeles de Diffusion avec Sautes, Comptes Rendus de l'Academie des Science de Paris Serie I (1993) 857–862.